INTRUSION DETECTION SYSTEM USING CUSTOMIZED RULES FOR SNORT

Manju[1], Shanmugasundaram Hariharan[2], M. Mahasree[1], Andraju Bhanu Prasad[2] and H.Venkateswara Reddy[2]

[1]Department of CSE, SRM Institute of Science and Technology, India
[2]Department of CSE, Vardhaman College of Engineering Hyderabad, India

ABSTRACT

These days the security provided by the computer systems is a big issue as it always has the threats of cyber-attacks like IP address spoofing, Denial of Service (DOS), token impersonation, etc. The security provided by the blue team operations tends to be costly if done in large firms as a large number of systems need to be protected against these attacks. This leads these firms to turn to less costly security configurations like IDS Suricata and IDS Snort. The main theme of the project is to improve the services provided by Snort which is a tool used in creating a vague defense against cyber-attacks like DDOS attacks which are done on both physical and network layers. These attacks in turn result in loss of extremely important data. The rules defined in this project will result in monitoring traffic, analyzing it, and taking appropriate action to not only stop the attack but also locate its source IP address. This whole process uses different tools other than Snort like Wireshark, Wazuh and Splunk. The product of this will result in not only the detection of the attack but also the source IP address of the machine on which the attack is initiated and completed. The end product of this research will result in sets of default rules for the Snort tool which will not only be able to provide better security than its previous versions but also be able to provide the user with the IP address of the attacker or the person conducting the attack. The system involves the integration of Wazuh with Snort tool in order to make it more efficient than IDS Suricata which is another intrusion detection system capable of detecting all these types of attacks as mentioned. Splunk is another tool used in this project which increases the firewall efficiency to pass the no. of bits to be scanned and the no. of bits scanned successfully. Wazuh is used in this system as it is the best choice for traffic monitoring and incident response than any other of its alternatives in the market. Since this system is used in firms which are known to handle big amounts of data and for this purpose, we use Splunk tool as it is very efficient in handling big amounts of data. Wireshark is used in this system in order to give the IDS automation in its capability to capture and report the malicious packets found during the network scan. All of this gives the IDS a capability of a low budget automated threat detection system. This paper gives complete guidelines for authors submitting papers for the AIRCC Journals.

KEYWORDS

Intrusion Detection System, Snort, Wireshark, Wazuh, Splunk, DDOS attack, Automation.

1. INTRODUCTION

The attacks faced today by the industry are known to have a disastrous effect on the data privacy and security of the companies, which is an important part of patent infringement prevention and employee data security [27]. The security provided by the big firms these days are effective in most cases but they tend to be more and more costly day by day. This leaves the small budget firms and companies unprotected or end up in trusting not suitable software systems. A solution to this can be provided by using the solution provided in this paper which uses an appropriate

approach for protecting the data by alerting the customer by performing a basic blue team operation scan. An Intrusion Detection System (IDS) is a system that monitors network traffic for suspicious activity and issues alerts when such activity is discovered. It is a software application that scans a network or a system for the harmful activity or policy breaching. Any malicious venture or violation is normally reported either to an administrator or collected centrally using a security information and event management (SIEM) system.

SIEM system integrates outputs from multiple sources and uses alarm filtering techniques to differentiate malicious activity from false alarms. Although intrusion detection systems monitor networks for potentially malicious activity, they are also disposed to false alarms [28, 29]. Hence, organizations need to fine-tune their IDS products when they first install them. It means properly setting up the intrusion detection systems to recognize what normal traffic on the network looks like as compared to malicious activity. Intrusion prevention systems also monitor network packets inbound the system to check the malicious activities involved in it and at once send the warning notifications. Intrusion detection system (IDS), an influential approach, is primarily implemented to detect abnormal activities in a target applications or computers. An IDS has two main methods [1], signature-based detection and anomaly-based detection. Signature-based detection is used to detect identified attacks using rule-based methods. Conversely, anomaly-based detection is utilized to detect both the known and unknown attacks by learning their behaviour. Intrusion prevention systems also monitor network packets inbound the system to check the malicious activities involved in it and at once send the warning notifications.

2. RELATED WORK

The main reason for the project to be created in the first place is that IDS Snort, the already existing version of the snort system, could not locate the IP address of the system attempting to attack. The already existing IDS is also effective against these attacks, but the time taken by the system to prevent damage from these attacks is much longer than the proposed system. The original design consisted of firewalls which protect the system from external threats. Those are two types, i.e., packet filtering router and application gateway [5]. Another type is a circuit-level gateway which typically relays TCP packets. Basic web security, like access level security and transaction level security, was also one of the core features of the existing design [4,5].

Issues in the existing system included training time which was a significant issue as setting up earlier IDS required a lot of training and setup time. During this time, the system was vulnerable as it needed the original security to be dismantled before setting up a new one [6]. Attack identification was also a big concern as it took hours to identify a simple brute force attack which was one of the low-level attacks.

In a recent study, cyber defence mechanism has become a more sophisticated and challenging approach for accurately detecting intrusion detection [25]. Data confidentiality, integrity, and availability need more attention and have been widely addressed in wireless-compatible networks [23, 24]. Numerous intrusion approaches are widely surveyed in literature to tackle computer security threats, which can be broadly classified into Signature-based Intrusion Detection Systems (SIDS) and Anomaly-based Intrusion Detection Systems (AIDS). This approach discusses several review comparisons, dataset illustrations, workflow, and evaluation mechanisms. The techniques widely investigated by attackers in detecting network traffic analysis and future research challenges to counter such methods for making a secure approach make the study attractive [21, 26].

Intrusion detection provides security not only for an individual entity but also for an organization as a whole. Several exciting studies for intrusion detection exist. One such approach is the use of

the ontology approach. The work addresses the problems that handle syntax and semantics, affecting the IDS system performance. SNORT rules were designed in a more challenging manner, and such design supports SNORT rule verification using OWL ontology. The function is to detect harmful traffic from the ontology design construed. The primary issue in this task would be the design of efficient detection and SNORT rule specification [7].

Similar other approaches, which were expressive in creating rules, were not only trivial tasks, especially for the self-defined rule but also needed to be more challenging in measuring the complexity of the network creation process for network monitoring. The work presented in the approach discusses a signature-based approach called Network Intrusion Detection Systems (NIDS). Most systems were designed for something other than modern high-speed network capability, which includes a choice for such high-speed networks, network flow-monitoring and Internet Protocol Flow Information Export (IPFIX) definition standard. There exist no current solutions that are even to handle payload in these flows. Recently, the concept of application layer HTTP flow has been extended to improve the version of the IPFIX-based Signature-based Intrusion Detection System (FIXIDS). The study measures the evaluation that could be of high deal with four times higher network data rates without drops than Snort while maintaining the same event detection rate. Furthermore, a substantial part of the data traffic can be outsourced to Fixids so that Snort can be relieved of a significant portion of rules and traffic. This increases the detection and data rates the overall security appliance can handle [24].

3. PROPOSED ARCHITECTURE

The architecture followed by the system is presented here and outlined in detail.

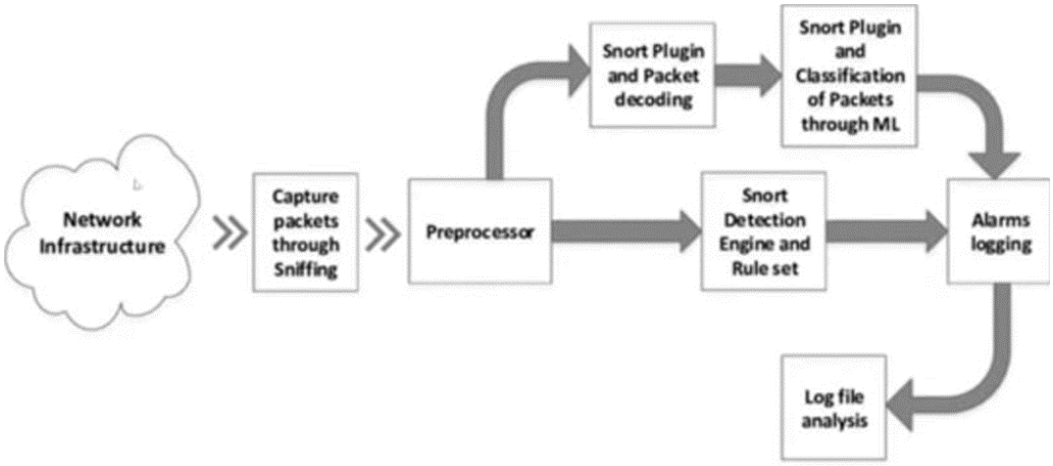

Figure 1. Proposed system architecture

Figure 1 shows why our approach is better than any other machine learning approach applied in the system in the usual case as we reach the same result of alarm logging in the end without allowing the project to make a leap for a high processor needing approach of machine learning [26].

The step of preprocessor is the step where Wireshark will decide to make the approach to snort tool in order for it to check for packet anomaly. The step of log file analysis is where our project will be applying the tools wazuh and Splunk.

4. INTRUSION DETECTION SYSTEM USING CUSTOMIZED RULES FOR SNORT

The tools involved in the project include Wireshark, Snort, Wazuh and Splunk. Each tool needs to be setup like the normal IDS configuration tool but the time taken to setup these tools is comparatively very low on an Ideal system which means that it means the minimum requirement for running it efficiently. For most of these tools there is no specific OS required but for Snort tool to run the experiments related to this paper and research community rules were imported which could be found on snort.org.

**Wireshark*-This tool is very essential part of the whole IDS as it does for the system the basic reconnaissance which detects the first crucial minutes of the attacks. Basically, what it does is, it scans the traffic of the machine on which the scan is performed and each packet is scanned for the time at which it made a transition, the type of security the communication channel is using and the encryption it uses for the packet, the type of protocol used, the layer in which the communication is taking place. All of this is done in real time as each packet passes through the scan.

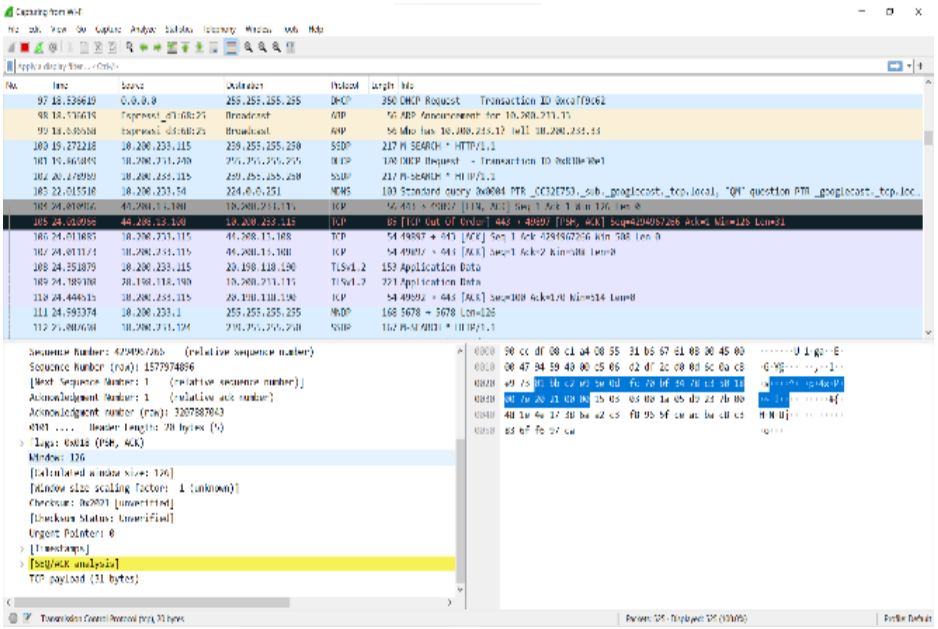

Figure 2. Wireshark tool showing packet configuration

The black highlighted packet in the Figure 2 is a suspicious packet as the tool has found an anomaly in the packet. Grey highlighted packet shown in the image is an analyzed packet in which the anomaly is found and dealt with. This shows a silent feature of the tool which is very helpful in our case and is one of the main reasons for using this tool which is that is automatically moves onto the next packet analyzation after completing the action on one packet.

**Snort*-This is the most important tool of our setup as it is responsible for incident response for the attack and also helps with the blue team effort for the first line of defense against the attack. It consists of rules which are divided into different categories such as community rules, user defined rules and etc. Out of these, community rules include most of the rules required for avoiding most of the attacks but most of the time due to large number of rules it takes a long time to analyze a single packet and during this time the attack has already caused damage to the system. For this reason, we are creating a particular rule set which decreases the time taken by the

IDS to analyze and take action regarding the packet. This is far more efficient for basic attacks like brute force attack and phishing attack.

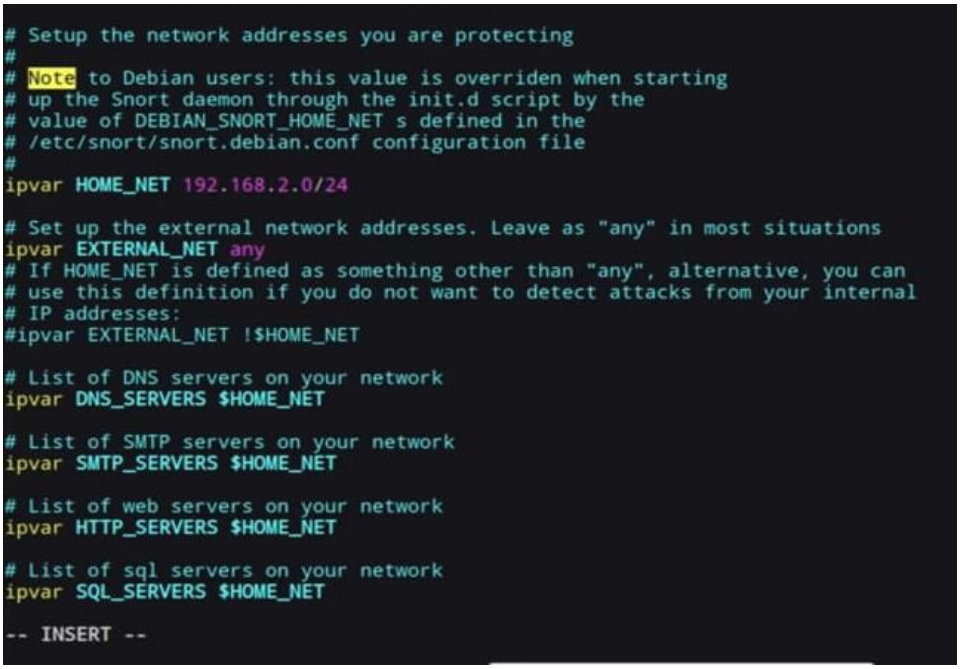

Figure 3. Rulesets

In the Figure 3 multiple rule sets for different servers are shown this is also another feature of snort that the rules can be defined for a particular section at a time which increases the time efficiency of the system. We can also add additional rule set by using “ipvar <name of the server to be applied to> $HOME_NET” command. Then comes the permissions provided by the user to the tool in order to protect the sections for privacy protection. As we know that there are read, write and execute permissions for the tools to be used for the files. This file permission is the tricky part for the configurations as it depends on user how much access he/she wants to give to the tool.

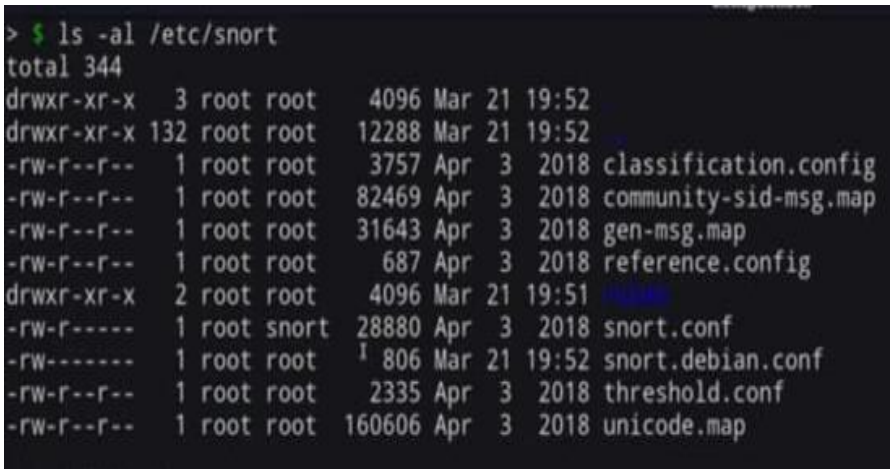

Figure 4. Configuration Settings

Figure 4 shows the different configurations done to the tool permissions in order to give the tool access to operate freely on the desired files. There can be different set of permissions to different sets of files depending on the user. Another benefit of using this tool is that it acts as a wall between any communication to the server and an external connection as shown in figure 5

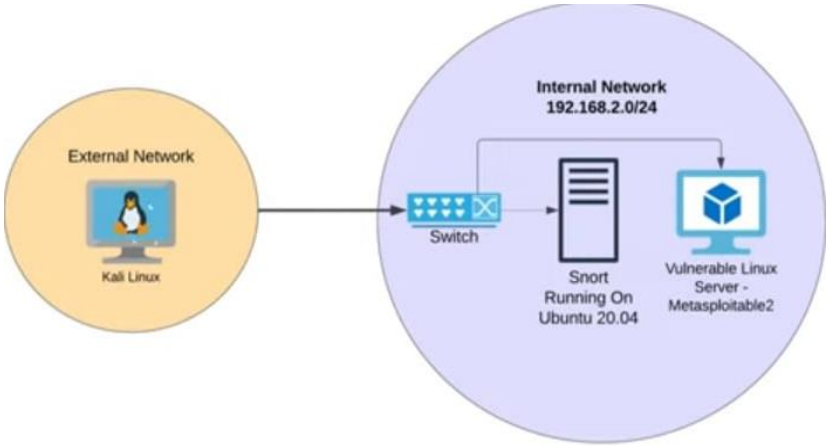

Figure 5. Snort framework

Then comes the part of exclusion of rules which can be done easily by giving the system the serial no. of the rules to be excluded. For example: If we want to disable rules from 578 to 696 which is set of rules for checking the admin access for the downloads folder for the system then we need to type "578,696s/^/#" and the rules for that part will be excluded as shown in Figure 6.

```
689 include $RULE_PATH/community-web-misc.rules
690 include $RULE_PATH/community-web-php.rules
691 include $RULE_PATH/community-sql-injection.rules
692 include $RULE_PATH/community-web-client.rules
693 include $RULE_PATH/community-web-dos.rules
694 include $RULE_PATH/community-web-iis.rules
695 include $RULE_PATH/community-web-misc.rules
696 include $RULE_PATH/community-web-php.rules
697
698
699 ##################################################
700 # Step #8: Customize your preprocessor and decoder alerts
701 # For more information, see README.decoder_preproc_rules
:578,696s/^/#
```

Figure 6. Exclusion of rules

Finally, the system will give a check of the rules applied to the system and give the details regarding which ones have been violated, which rules have found an anomaly in the system, files which are out of order and the packet which was received was a malicious one or not (presented in Figure 7). This all information is given in a single report format.

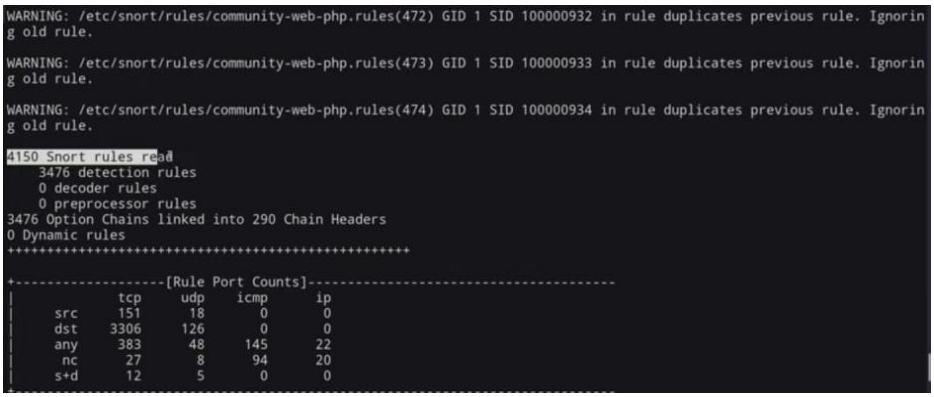

Figure 7. Identification of malicious packet

After this we come to designing of rule sets of our own which is a process that might differ from user not user as they not only design the rules but also check for violations with other rules. There is a format in which the rules need to be designed which is shown in Figure 8.

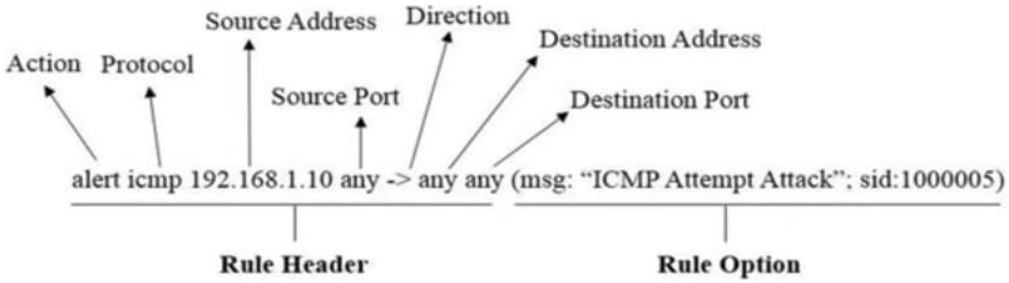

Figure. 8 Proposed rule format

After the rule has been designed it is really hard to check for the violations that this rule has with other existing rules which can be checked while applying those to the system. For the ease of creating the rules and checking them at the same time we used Snorpy tool, as shown in figure 9, which is easy to use as well as customization of the rules is also done a lot faster than the normal speed.

Figure 9. Snorpy tool for checking rules

Snorpy tool has different protocols to customize from along with each field defined clearly for rule generation and the final rule is shown in the base box for finally checking the rule before it is applied. After the rules has been created, we need to insert it which can be done by addition process of rules in snort tool which is given as screenshot in Figure 10.

```
include $RULE_PATH/local.rules

# The include files commented below have been disabled
# because they are not available in the stock Debian
# rules. If you install the Sourcefire VRT please make
# sure you re-enable them again:

#include $RULE_PATH/app-detect.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/bad-traffic.rules
#include $RULE_PATH/blacklist.rules
#include $RULE_PATH/botnet-cnc.rules
#include $RULE_PATH/browser-chrome.rules
#include $RULE_PATH/browser-firefox.rules
#include $RULE_PATH/browser-ie.rules
#include $RULE_PATH/browser-other.rules
#include $RULE_PATH/browser-plugins.rules
#include $RULE_PATH/browser-webkit.rules
include $RULE_PATH/chat.rules
#include $RULE_PATH/content-replace.rules
-- INSERT --
```

Figure 10. Addition of rules.

In the final hierarchy snort is placed along with the IDS in the following place, as depicted in figure 11., i.e., behind the firewall as it will be looking at traffic and anomaly after it has passed through the firewall protection.

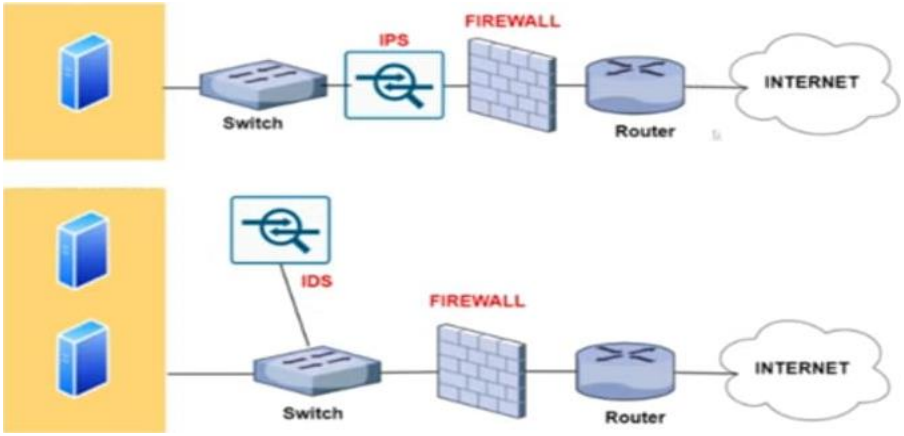

Figure 11. Position of Snort with IDS

The rules created as the result of this process are:

```
alert icmp any any -> $HOME_NET any (msg:"ICMP Ping Detected"; sid:100001; rev:1;)
alert tcp any any -> $HOME_NET 22 (msg:"SSH Authentication Attempt"; sid:100002; rev:1;)
alert tcp any any -> 192.168.2.157 21 ( msg:"FTP Authentication Attempt On MS2"; sid:100003; rev:1; )
```

Figure 12. Result of Snort placement – detection of attacks

These rules operate in the TCP and ICMP layer of the OSI model, will look for ICMP ping attack, SSH authentication attempt and FTP authentication attempt. The messages for detecting each of these attacks will also be displayed as shown in Figure 12. As for the results for attempting the attack and seeing if this works or not are also recorded in Figure 13.

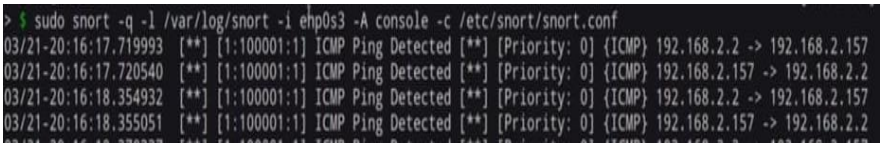

Figure 13. Attempts on breaking the system

Figure 14 shown is the result after attempting to attack by ICMP ping impersonation method.

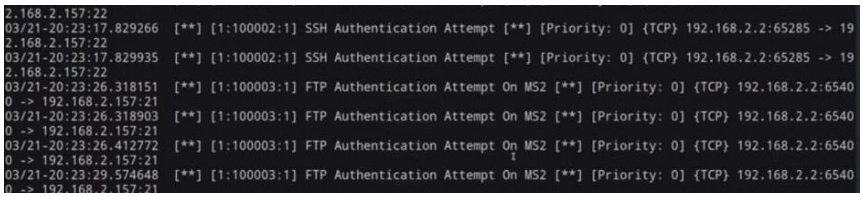

Figure 14. Result of attempting the attacks

This image shows a SSH authentication attempt on the system which is one of the basic steps of brute force attacks.

**Wazuh*-This tool is used for checking for other attacks as it looks out for the session time outs and other attacks like ping impersonation which cannot be detected by using only snort tool. It is shown in Figure 15.

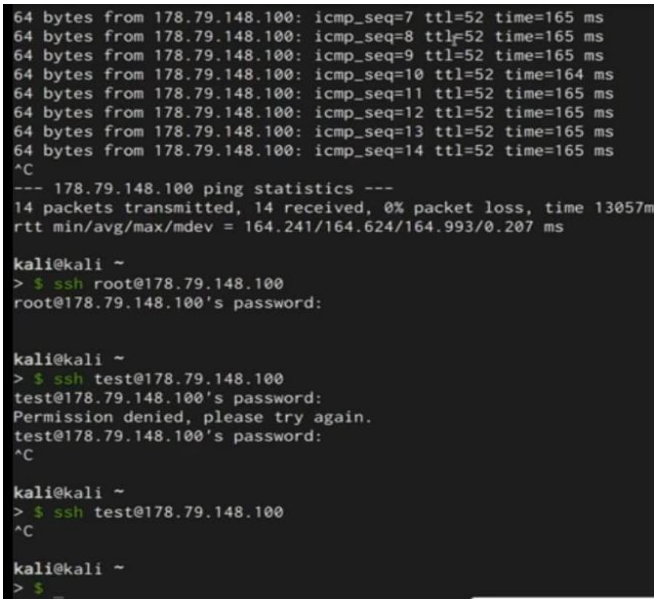

Figure 15. Checking for impersonation using Wazuh tool

It has the role of granting permission to the snort tool and detecting the level of security breach faced by the system which is described in Figure 16.

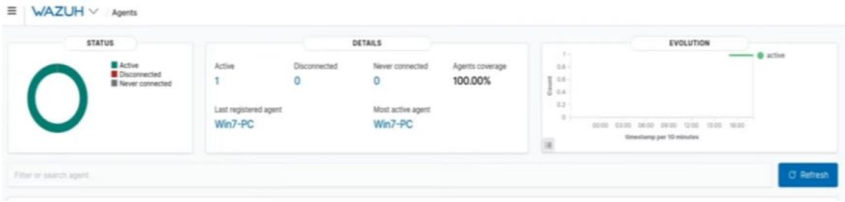

Figure 16. Granting permission to snort tool

It works in a report-based format as it gives report on the damage faced by the files, and level of vulnerability that the system has at the time (as presented in Figure 17).

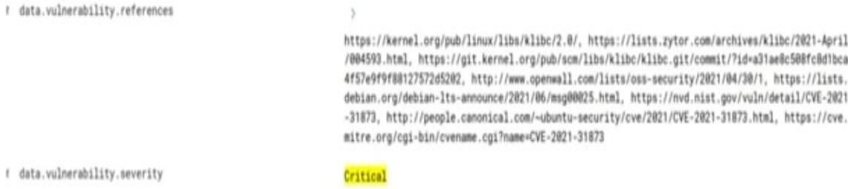

Figure 17. Vulnerability report generation

The image if from the time when the system detected the attack and labeled the critical threat level. As far as ping detection goes it can be customized in the base code in the following method as in Figure 18 and the system will show the detection if it finds an anomaly as shown in Figure 19.

Figure 18. Labeling the critical threat level with time

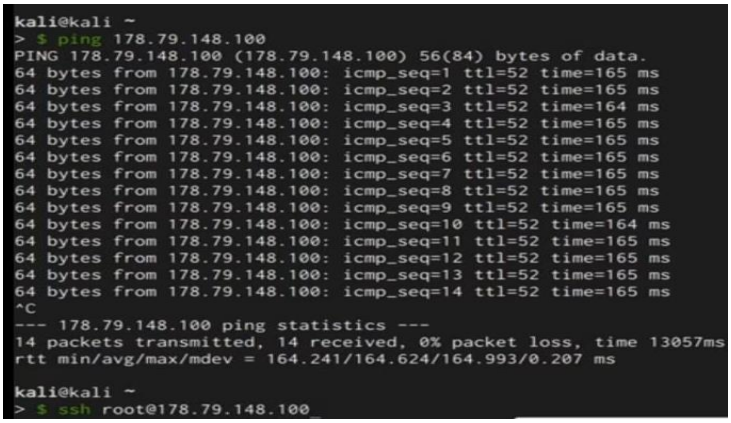

Figure 19. Anomaly detection

**Splunk*-This tool is responsible for interpreting the result given by Snort tool and giving it in a report format. It is essential as the result given by snort rules is not in human readable format as highlighted in figure 20.

Figure 20. Report Generation using Splunk tool

This result given by Snort and Wazuh integrating is then easily interpreted by Splunk and given in the format readable by us humans i.e. in Figure 21.

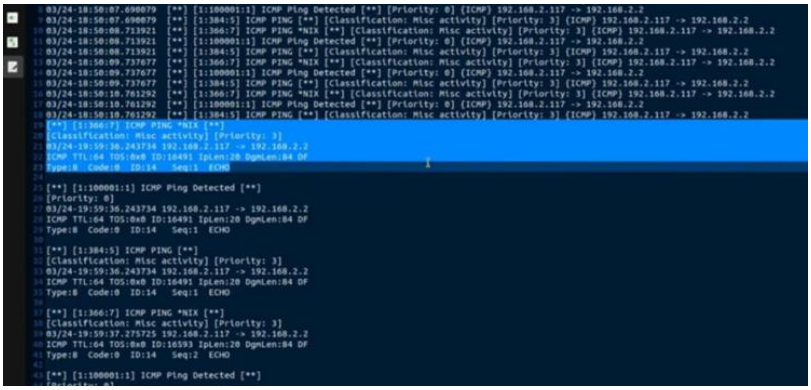

Figure 21. Result of Snort and Wazuh integration

Finally, the report of the attack is given by Splunk in the following report format as shown in figure 22.

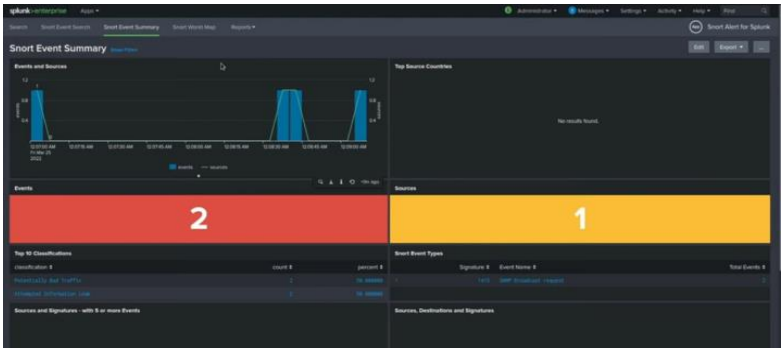

Figure 22. Report about attacks by Splunk

It gives us a side advantage of graph which helps us understand better the whole timeline of events which makes incident response with our system better than any other on the market already present.

5. CONCLUSION AND FUTURE WORK

The final conclusion drawn from the project is that the IDS system is a vast field and there is a scope for a lot of approaches in the field. Our system eliminated the need for a machine learning algorithm for basic attacks like brute force attack, token impersonation attack, ssh privilege escalation attack. These attacks don't require a lot of effort which is put by other high-level algorithms that use machine learning as they could be handled in a matter of few hours than a whole 7 to 8 hours. This project is not a method or approach of trying to remove focus from developing those machine learning projects but is a quicker way of handling small scale attacks than so that those large-scale algorithms could be put up to use in other large-scale attacks which cannot be solved by our system.

The IDS used a whole of 4 tools which are Wireshark, Snort, Wazuh and Splunk. Out of these Wireshark produced the result of initial reconnaissance by analyzing the traffic and reporting on the malicious packet. Snort used the set of rules from the user and the community section to find the anomaly in the packet given by Wireshark. Side by side snort is integrated with wazuh for checking ping impersonation attempts which might slow snort operations and also allows snort permissions to conduct changes in the packet. Then after analyzing the result wazuh adds its results to snort report and gives a human unreadable result. This result is then sent to Splunk which analyzes it and gives a human readable result along with the incident report which was generated from the result.The whole process took a time of 5 hours which is a lot less than any known algorithm in the market. The usual time taken by the process is 7 to 8 hours which gives our method a total of approximately 3 hours of advantage which is helpful in incident response.

ACKNOWLEDGEMENTS

The authors would like to thank anonymous reviewers for their valuable comments.

REFERENCES

[1] D. Fadhilah and M. Ihsan Marzuki, et al. (2020), Performance Analysis of IDS Snort and IDS Suricata with Many-Core Processor in Virtual Machines Against Dos/DDoS Attacks, 2020 2nd International Conference on Broadband Communications, Wireless Sensors and Powering (BCWSP)

[2] R. Abubakar, A. Aldegheishem, et al. (2020), An Effective Mechanism to Mitigate Real-Time DDoS Attack, IEEE Access (Volume: 8)

[3] SanatSarada, Roland Rieke (2022) Decision Tree-Based Rule Derivation for Intrusion Detection in Safety-Critical Automotive Systems

[4] PavolZavarsky (2020) Deep Packet Inspection in Industrial Automation Control System to Mitigate Attacks Exploiting Modbus/TCP Vulnerabilities

[5] Denis Atanasov, KirilKassive (2020), Intrusion Detection System Model Implementation against DDOS attacks

[6] Raja Majid Ali Ujjan, KeshavDahal (2020), Snort-Based Collaborative Intrusion Detection System Using Blockchain in SDN

[7] A. Khurat and W. Sawangphol, "An Ontology for SNORT Rule," 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE), Chonburi, Thailand, 2019, pp. 49-55, doi: 10.1109/JCSSE.2019.8864190.

[8] Imran Shafi, Atif Ali (2020), Performance Enhancement of Snort IDS through Kernel Modification

[9] Ravi Shankar, Aman Singh (2022), Analysis of Network Attacks at Data Link Layer and its Mitigation

[10] H. Premkumar and A. P Patil (2022), JARVIS, An Intelligent Network Intrusion Detection and Prevention System

[11] A. Alsaleh and W. Binsaeedan, "The Influence of Salp Swarm Algorithm-Based Feature Selection on Network Anomaly Intrusion Detection",Journal of Network Security, Aug 2021

[12] G. Pu, L.Wang, J. Shen and F. Dong, "A Hybrid Unsupervised Clustering-Based Anomaly Detection Method", April 2021.

[13] A.Mezina, R. Burget and C. M. T. Conzalez, "Network Anomaly Detection with Temporal Convolutional Network and U-Net Model", Journal of Signals and Communication, Oct 2021.

[14] B. Adhi Tama and L. Nkenyereye, "An Enhanced Anomaly Detection in Web Traffic Using a Stack of Classifier Ensemble", 2020.

[15] D. Fadhilah and M. I. Marzuki, "Performance Analysis of IDS Snort and IDS Suricata with Many-Core Processor in Virtual Machines Against Dos/DDoS Attacks," 2020 2nd International Conference on Broadband Communications, Wireless Sensors and Powering (BCWSP), Yogyakarta, Indonesia, 2020, pp. 157-162, doi: 10.1109/BCWSP50066.2020.9249449.

[16] E. Tufan, C. Tezcan and C. Acartürk, "Anomaly-Based Intrusion Detection by Machine Learning: A Case Study on Probing Attacks to an Institutional Network", IEEE Access, 2021.

[17] W. Alhakami, A. Alharbi, S. Bourouis, R.Alroobaea and N. Bouguila, "Network Anomaly Intrusion Detection Using a Nonparametric Bayesian Approach and Feature Selection", January 2019

[18] I. A. Khan, D. Pi, Z. U. Khan, Y. Hussain and A. Nawaz, "HML-IDS: A Hybrid-Multilevel Anomaly Prediction Approach for Intrusion Detection in SCADA Systems", IEEE Access, July 2019.

[19] Y. He and J. Zhao, "Temporal Convolutional Networks for Anomaly Detection in Time Series", Journal of Physics Conference Series", June 2019.

[20] A. A. Tama and L. Nkenyeyeye , "An Enhanced Anomaly Detection in Web Traffic Using a Stack of Classifier Ensemble", Feb 2020.

[21] A. Khraisat, I. Gondal and P. Vamplew et al, "Survey of intrusion detection systems: techniques, datasets and challenges", (2019). https://doi.org/10.1186/s42400-019-0038-7.

[22] Z. Zhou, Z. Chen, T. Zhou and X. Guan, "The study on network intrusion detection system of Snort," 2010 International Conference on Networking and Digital Society, Wenzhou, China, 2010, pp. 194-196, doi: 10.1109/ICNDS.2010.5479341.

[23] T. Chand and B. Sharma, "HRCCTP: a hybrid reliable and congestion control transport protocol for wireless sensor networks", IEEE sensors, pp. 1-4, 2015.

[24] R. Dogra, S. Rani, B Sharma and S. Verma, "Essence of scalability in wireless sensor network for smart city applications", In IOP Conference Series: Materials Science and Engineering, vol. 1022, No. 1, p. 012094). IOP Publishing, 2021.

[25] S. Srujana, P. Sreeja, G. Swetha and H. Shanmugasundaram, "Cutting Edge Technologies for Improved Cybersecurity Model: A Survey," 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2022, pp. 1392-1396, doi: 10.1109/ICAAIC53929.2022.9793228.

[26] D. Dhanalakshmi, N. D. Rani, K. Pendam, S. Hariharan, V. Kukreja and P. Jayakshata, "Machine Learning based Intelligent Cyberbullying Avoidance System," 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2023, pp. 1594-1597, doi: 10.1109/ICSCSS57650.2023.10169376.

[27] B. N, P. K, M. S, H. S, V. K. M and V. MRM, "A Novel Framework for Cyber Security Attacks on Cloud-Based Services," 2022 Fourth International Conference on Cognitive Computing and Information Processing (CCIP), Bengaluru, India, 2022, pp. 1-4, doi: 10.1109/CCIP57447.2022.10058673.

[28] A.H. Wheeb, "Performance Analysis of VoIP in Wireless Networks," International Journal of Computer Networks and Wireless Communications (IJCNWC), vol. 7, no. 4, pp. 1-5, 2017.

[29] D. N. Kanellopoulos and A. H. Wheeb, "Simulated Performance of TFRC, DCCP, SCTP, and UDP Protocols Over Wired Networks," Int. J. Interdiscip. Telecommun. Netw., vol. 12, no. 4, pp. 88–103, 2020, doi: 10.4018/ijitn.2020100107.

AUTHORS

Dr. A. Manju is working as Assistant Professor in the Department of CSE at SRMIST, Ramapuram campus. She holds a Ph. D degree in the field of Computer Vision and Video Analytics from Saveetha Institute of Medical and Technical Sciences (SIMATS) since 2022. She has 16 years of teaching experience with good programming skills. She is a certified EMC Academic Associate in Data Science & Big Data Analytics and also certified in various courses from Coursera and NPTEL. She has published articles in National and International Journals, Conferences and Symposiums.

Dr. S. Hariharan received his B.E degree specialized in Computer Science and Engineering from Madurai Kammaraj University, Madurai, India in 2002, M.E degree specialized in the field of Computer Science and Engineering from Anna University, Chennai, India in 2004 and Ph.D degree in the area of Information Retrieval from Anna University, Chennai, India in the year 2010. He is a Senior member of IEEE and member of several other professional societies. He has 19 years of experience in teaching. Currently he is working as Professor in Department of Computer Science and Engineering, Vardhaman College of Engineering, Hyderabad, India. His research interests include Information Retrieval, Data mining, Opinion Mining, Web mining. He has to his credit several papers in referred journals and conferences. He also serves as editorial board member and as program committee member for several international journals and conferences.

Dr. M. Mahasree is working as Assistant Professor in the Department of CSE at SRMIST, Ramapuram campus. She holds a Ph.D degree in Computer Science and Engineering from Annamalai University since 2022. She has completed several certified courses from NPTEL. She has presented papers in National and International Conferences. She also has published articles, book chapters in International Journals. Her area of research includes computer vision, data security and deep learning

Andraju Bhanu Prasad M.E (CSE) from Sathyabama University, Chennai, B.Tech (CSE) from SVCET JNTUH, Hyderabad. Having 17 years of experience in Academics and currently he is an Associate Professor at Vardhaman College of Engineering, Hyderabad, research areas are Data science, Machine Learning and Deep Learning. Currenlty he is pursuing his doctoral degree in the area of machine learnig. He has published several research articles in refereed journals and conferences. He also authored two books in the cybersecurity and big data.

Dr. H. Venkateswara Reddy is a Doctoral Fellow from JNTUH, Hyderabad, India and working at this institute since 2002 at various capacities as Professor and Head of the Department of CSE at present doing the services as Control of the Examinations. Worked as deeper investigation of Study on Rough Sets Theory. He has published more than 35 research articles in international and national journals and conferences, one Indian patent got published titled as “Face Detection and Recognition using SVM and HOT Technologies” and handled two DST projects. He is also a Principle Investigator of a research project approved by DST is carrying out in the field of Cognitive science with the title “Understanding Bisociation capabilities in Indian engineering students” amount of 31 lackhs and my second DST project titled as “Rural Women Technological Park” worked as a co-investigator. Currently he is guiding Ph.D Research Scholars affiliated to JNTUH and Annamalai University.